



**SAVONIA**

■ OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO  
TEKNIIKAN JA LIIKENTEEN ALA

# POWERSHELLIN HYÖDYN- TÄMINEN ENFO SERVICE DESKISSÄ

TEKIJÄ/T: Janne Kauppinen

Koulutusala Tekniikan ja liikenteen ala	
Koulutusohjelma Tietotekniikan koulutusohjelma	
Työn tekijä(t) Janne Kauppinen	
Työn nimi PowerShellin hyödyntäminen Enfo Service Deskissä	
Päiväys 7.5.2015	Sivumäärä/Liitteet 31
Ohjaaja(t) lehtori Jussi Koistinen, lehtori Keijo Kuosmanen	
Toimeksiantaja/Yhteistyökumppani(t) Enfo Oyj	
<p>Tiivistelmä</p> <p>Opinnäytetyössä toteutettiin useita PowerShell-skriptejä Active Directory-, Exchange-, Lync- sekä hakemisto-objektien hallitsemiseksi. Tehdyistä skripteistä koottiin funktiokirjasto, joka sisälsi noin 30 valmista funktiota. Työ tehtiin Enfo Oyj:n Service Desk -yksikölle tehostamaan sekä helpottamaan työntekoa.</p> <p>Työ aloitettiin perehtymällä PowerShellin käyttömahdollisuuksiin. Tämän jälkeen selvitettiin, mihin tarkoituksiin Service Deskissä sillä hetkellä käytettiin PowerShell-skriptejä. Selvityksen perusteella päätettiin, mitä PowerShell-skriptejä luodaan ja mihin tarkoitukseen. Enfo Service Deskin toimintamallit olivat tulleet tutuksi yli vuoden työssäolon aikana, mikä helpotti päätöksentekoa. Lopuksi toteutettiin suunnitellut skriptit sekä funktiokirjasto.</p> <p>Työn lopputulokseksi tuli noin 30 skriptin funktiokirjasto ja noin 10 valmista skripti-kokonaisuutta. Työssä luotujen skriptien avulla voidaan toteuttaa usean Enfo Oyj:n asiakasyrityksen Active Directory-, Exchange-, Lync- sekä hakemisto-objektien hallinta. Työn lopputuotteina Enfo Oyj:lle luovutettiin työssä luodut skriptit sekä dokumentit skriptien käyttämiseksi.</p>	
Avainsanat Service Desk, PowerShell, Active Directory, Windows Server	

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Janne Kauppinen			
Title of Thesis Exploitation PowerShell-Scripts in Enfo Service Desk			
Date	7 May 2015	Pages/Appendices	31
Supervisor(s) Mr. Jussi Koistinen, Lecturer and Mr. Keijo Kuosmanen, Lecturer			
Client Organisation /Partners Enfo Oyj			
<p>Abstract</p> <p>The purpose of this thesis was to create a variety of PowerShell scripts to manage different Active Directory-, Exchange-, Lync- and directory objects. The scripts that were done were compiled to a function-library that contained about 30 functions. The thesis was made for Enfo Oyj's Service Desk unit to increase the employee's efficiency and to facilitate their work.</p> <p>The work was started by exploring the possibilities to use PowerShell. After that the purpose for which Service Desk currently uses PowerShell scripts was examined. After research it was decided what PowerShell scripts will be created and for what purpose. Enfo Service Desk policies had already become familiar during about one year's employment which facilitated decision-making. Finally, the planned scripts and function-library were made.</p> <p>The final result was a function-library that contained around 30 functions and about 10 completed scripts. The scripts that were made in this thesis can be used to implement the management of a number of Enfo Oyj's clients Active Directory, Exchange, Lync and directory objects. The final products that were handed to Enfo Oyj were the function-library and the documents on how to use these scripts.</p>			
Keywords Service Desk, PowerShell, Active Directory, Windows Server			

## ESIPUHE

Tahdon kiittää Enfo Oyj:tä opinnäytetyön aiheesta. Kiitos myös toimeksiantajan ohjaajalle järjestelmäasiantuntija Tomi Monoselle sekä ohjaavalle opettajalle lehtori Jussi Koistiselle työn ohjauksesta sekä saamistani neuvoista työn aikana.

Kuopiossa 26.4.2015

Janne Kauppinen

## SISÄLTÖ

LYHENTEET JA MÄÄRITELMÄT .....	7
1 JOHDANTO .....	8
2 TYÖN TAUSTAT .....	9
2.1 Toimeksiantaja .....	9
2.2 Työn tausta .....	9
3 SERVICE DESK .....	10
3.1 Palvelut ja toiminta .....	10
3.2 Työpyynnöt ja niiden käsittely .....	10
4 WINDOWS-PALVELINYMPÄRISTÖ .....	12
4.1 Windows Server .....	12
4.2 Käytettävät palvelinversiot .....	13
4.2.1 Windows Server 2008 .....	13
4.2.2 Windows Server 2008 R2 .....	13
4.2.3 Windows Server 2012 .....	13
4.2.4 Windows Server 2012 R2 .....	14
4.2.5 Microsoft Exchange - ja Lync Server .....	14
4.3 Active Directory .....	14
5 POWERSHELL .....	17
5.1 Versiot .....	17
5.2 Komennot .....	18
5.2.1 Cmdlet-komennot .....	18
5.2.2 Shell-Funktiokomennot .....	19
5.2.3 Skriptikomennot .....	19
5.2.4 Natiivit Windows-komennot .....	19
5.3 Putkitus .....	20
6 TEKNINEN TOTEUTUS .....	21
6.1 Lähtökohdat ja vaatimukset .....	21
6.2 Suunnittelu .....	21
6.3 Toteutus ja testaus .....	21
6.3.1 Tunnusten poistaminen .....	22
6.3.2 Tunnusten luonti .....	23

6.3.3	Sähköpostilaatikon luonti resursseille.....	28
6.3.4	Testaus .....	28
7	YHTEENVETO.....	29
	LÄHTEET .....	30

## LYHENTEET JA MÄÄRITELMÄT

AD = Active Directory, Microsoftin kehittämä hierarkkinen käyttäjätietokanta ja hakemistopalvelu

CMDLETS = Command-Let, PowerShellin sisäänrakennettuja komentoja

Funktio = Ohjelmapätkä jota kutsutaan sen nimellä ja parametreilla

ISE = Integrated Scripting Enviroment, PowerShellin skriptienhallintatyökalu

OU = Organisation Unit, Hakemistoja, joiden avulla toimialueen objektit voidaan ryhmitellä järkevällä tavalla

Parametri = Komennolle suorituksen yhteydessä välitettävä arvo

PowerShell = Microsoftin kehittämä olio-pohjainen komentotulkki

Skripti = Sovelluksen suorittamisen aikana tulkattava komentosarja

## 1 JOHDANTO

Tämä opinnäytetyö toteutetaan Enfo Oyj:lle Service Desk -yksikön käyttöön. Työn tavoitteena on luoda Powershell-funktiokirjasto, joka sisältää Service Desk työntekijän työtä tehostavia ja helpottavia skriptifunktioita. Tällaisia toimintoja ovat muun muassa erilaisten Active Directory- sekä Exchange-objektien hallinta.

Enfo Oyj on pohjoismaissa vaikuttava IT-palvelutalo, joka tarjoaa asiakasyrityksilleen erilaisia IT- ja talousprosessien ulkoistuspalveluita sekä konsultointipalveluja. Service Desk toimii Enfon asiakasrajapinnassa. Enfon pääkonttori sijaitsee Kuopiossa.

PowerShell on Microsoftin kehittämä oliopohjainen komentotulkki, joka on tarkoitettu ylläpitäjille erilaisten toimintojen hallintaan sekä tehtävien automatisointiin.

Työn teoriaosuudessa esitellään toimeksiantaja ja työn taustat sekä kerrotaan Enfo Service Deskin toiminnasta. Tämän jälkeen perehdytään työssä käytettäviin palvelinkäyttöjärjestelmiin, Active Directoryyn ja sen rakenteeseen sekä PowerShelliin ja tärkeimpiin komentotyyppeihin. Käytännön osuudessa kerrotaan työssä luoduista PowerShell-skripteistä ja käydään läpi miten ja miksi skriptit toteutettiin.



## 2 TYÖN TAUSTAT

### 2.1 Toimeksiantaja

Enfo Oyj on IT-palvelutalo, joka tarjoaa asiakasyrityksilleen IT-ulkoistusta, talousprosessien ulkoistuspalveluita sekä IT-konsultointipalveluja pohjoismaissa. Enfon suurimmat toimipaikat sijaitsevat Suomessa ja Ruotsissa. Yritys on perustettu vuonna 1964 nimellä Tietosavo Oy, mutta vuonna 2001 nimeksi vaihtui Enfo Oyj. Enfo työllistää noin 800 tietotekniikka-alan ammattilaista. (Enfo Oyj 2014.)

### 2.2 Työn tausta

Opinnäytetyötä edelsi kolme harjoittelujaksoa Enfo Oyj:ssä sekä jo yli vuoden mittainen työsopimus Service Deskissä. Tänä aikana olin saanut hyvän kuvan Enfon toiminnasta sekä Service Deskissä työskentelystä, joten opinnäytetyön tekeminen samaan yritykseen tuntui luontevalta.

Halusin aloittaa opinnäytetyön tekemisen heti viimeisen harjoittelujakson jälkeen, joten kysyin mahdollisia opinnäytetyön aiheita silloiselta harjoittelun ohjaajaltani Sauli Rynnäseltä ja hän ehdotti aiheeksi *PowerShellin hyödyntäminen Service Deskissä ja PowerShell-työkalupakin luonti*. Aihe tuntui kiinnostavalta, sillä siitä voisi olla paljon hyötyä käytännön työssä Service Deskissä ja en ollut aiemmin käyttänyt PowerShellia.

Service Deskin työhön kuuluu muun muassa asiakkaiden tunnusten, sähköpostilaatikoiden sekä postituslistojen luonti ja poisto. Nämä toimet voivat viedä useamman minuutin Service Desk -työntekijän työaika, mikäli niiden toteuttamiseen ei käytetä PowerShell -skriptejä. PowerShell -skripteillä asiakkaan käyttäjätunnukset voidaan luoda erittäin nopeasti antamalla vain vaaditut parametrit, kuten esimerkiksi nimi ja käyttäjätunnus. Service Deskissä on olemassa jo yksittäisille asiakasyrityksille luotuja skriptejä, mutta näitäkin skriptejä pitäisi pystyä hyödyntämään laajemmin useamman asiakkaalla.

### 3 SERVICE DESK

Enfo Service Desk toimii Kuopiossa Viestikadulla. Lisäksi Puolassa Katowicessa sekä Ruotsissa Karlskronassa on Service Deskin toimipisteet. Service Deskissä työskentelee yhteensä 52 työntekijää.

#### 3.1 Palvelut ja toiminta

Service Desk tarjoaa asiakasyrityksilleen erilaisia ensimmäisen tason tukipalveluita, kuten järjestelmätuki, ohjelmistotuki, käyttöoikeustuki sekä käyttäjätuki. Service Desk toimii asiakkaan ensimmäisenä yhteydenottopisteenä.

Service Deskin tärkein tehtävä on kirjata asiakkaan ilmoittama tukipyyntö sekä luoda asiakkaalle tunne, että hänen pyyntönsä tai ongelmansa on huomioitu ja että se käsitellään ajallaan. Service Deskin motto onkin ”Jokainen asiakas, joka on meihin yhteydessä, saa positiivisen palvelukokemuksen”.

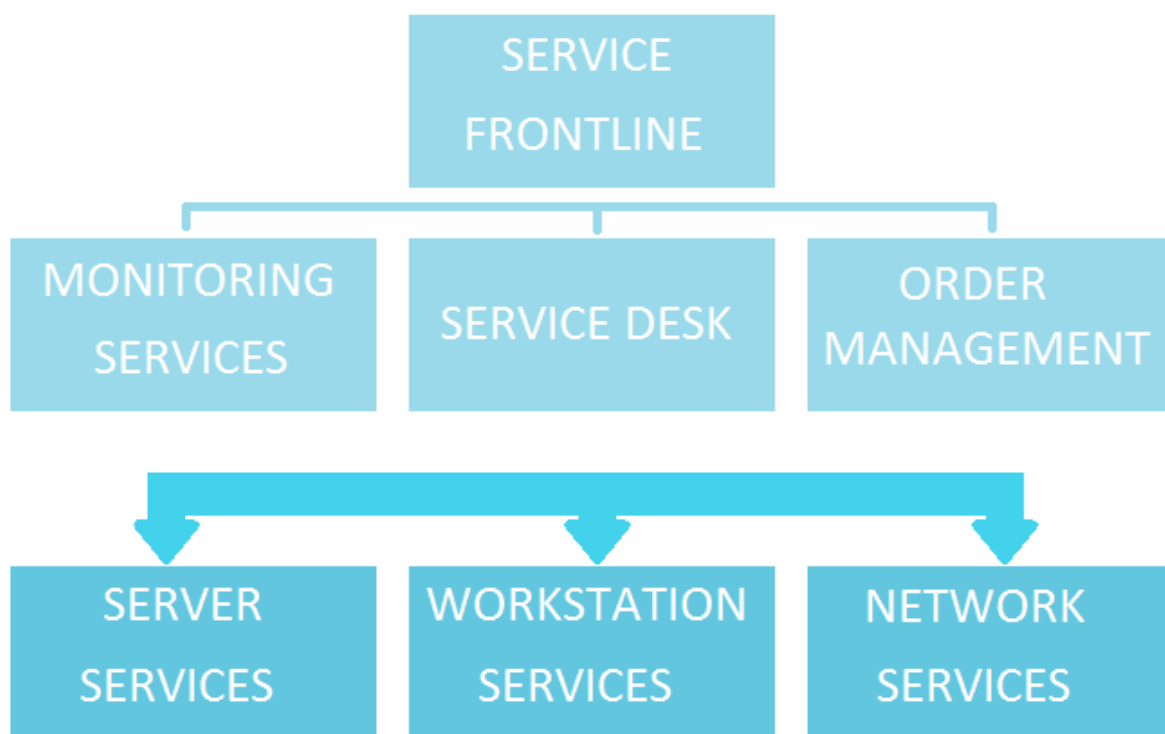
#### 3.2 Työpyynnöt ja niiden käsittely

Service Desk käsittelee monenlaisia työpyyntöjä. Service Desk luokittelee työpyynnöt joko palvelupyynnöiksi (Request) tai ongelmatilanteiksi (Incident).

Palvelupyynnöt ovat pyyntöjä, joissa asiakas haluaa esimerkiksi neuvoja, käyttöoikeuksia tai tukea jonkin ohjelman käyttöön. Esimerkkinä asiakas ilmoittaa työntekijän irtisanotuksi, jolloin irtisanotun henkilön tunnukset ja oikeudet suljetaan.

Ongelmatilanteiksi luokitellaan työpyynnöt, joissa häiriö tai ongelma haittaa asiakkaan työntekoa tai palvelun laatua. Esimerkiksi asiakkaan työn tekeminen on estynyt, koska työasema ei käynnisty.

Asiakas voi ottaa yhteyttä Service Deskiin puhelimella, sähköpostilla tai oman järjestelmänsä kautta. Asiakkaan tukipyynnöstä luodaan tikettijärjestelmään tiketti, joka kategorisoidaan joko palvelupyynnöksi tai ongelmatilanteeksi. Lähtökohtaisesti puhelimitse tulleet työpyynnöt yritetään ratkaista puhelun aikana, mutta mikäli työpyyntöön ei puhelun aikana syystä tai toisesta saada ratkaisua tai sen ratkaiseminen Service Deskissä ei ole mahdollista, se välitetään (eli eskaloidaan) tutkittavaksi toisille ryhmille. Esimerkiksi asiakas pyytää muutosta palvelusopimukseen, jolloin työpyynnöstä kirjataan Request ja se eskaloidaan palvelusopimuksista vastaavalle ryhmälle.



KUVA 1. Service Frontline sekä työpyyntöjen eskalointiryhmät (Kauppinen 2015-03-10.)

Kuvassa 1 on esitelty Enfo Service Frontlinen rakenne sekä työpyyntöjen eskalointiryhmät. Service Frontline koostuu Monitoring Servicestä, joka hallinnoi ja valvoo palvelimia, Service Deskistä sekä Order Managementista eli tilauksienhallinnasta, joka hoitaa laitteiden tilaukset.

Mikäli työpyyntöä ei saada ratkaistua Service Deskissä, se eskaloidaan ongelman perusteella joko Server Services-(palvelinten hallinta), Workstation Services-(työasemien hallinta) tai Network Services (verkon hallinta) -ryhmälle.

Työpyynnön etenemisestä sekä ratkaisusta kuuluu ilmoittaa asiakkaalle. Kun työpyyntö on ratkaistu, sen tilaksi vaihdetaan suljettu, mikäli työpyyntö on Request, tai ratkaistu, jos työpyyntö on Incident ja sille kirjoitetaan ratkaisu, minkä jälkeen järjestelmä lähettää automaattisen ilmoituksen asiakkaan sähköpostiin.

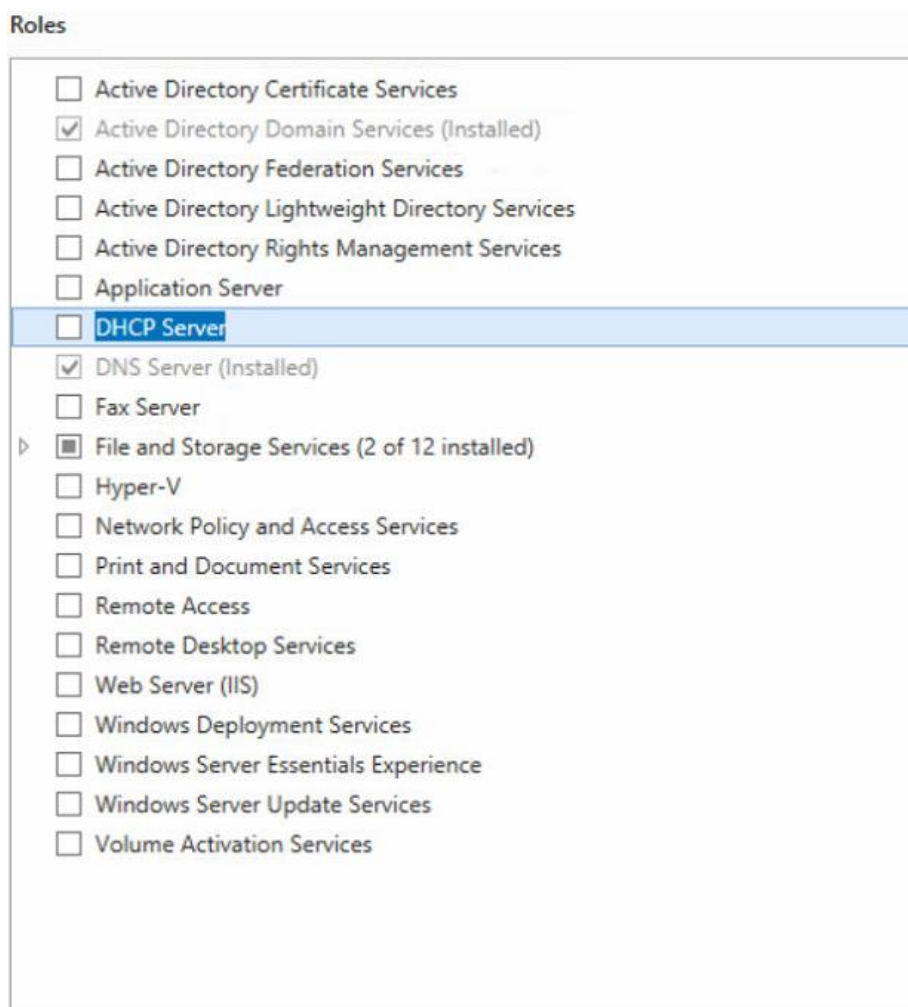
## 4 WINDOWS-PALVELINYMPÄRISTÖ

Tässä luvussa käsitellään työssä käytettäviä palvelinympäristöjä sekä Active Directory -hakemistopalvelua, koska eri palvelinympäristöjen välillä on eroja muun muassa käytettävissä komennoissa.

### 4.1 Windows Server

Windows Server on Microsoftin palvelinhallintaan tarkoitettu käyttöjärjestelmä. Sitä käytetään muun muassa yrityksen toimialueen työasemien ja käyttäjien hallintaan. Ensimmäinen Windows Server -käyttöjärjestelmä oli Windows Server 2000, joka julkaistiin vuoden 2000 helmikuussa. Se perustui Windows NT 5.0 -käyttöjärjestelmään ja oli viimeinen käyttöliittymältään Windows NT -tyylinen käyttöjärjestelmä. (Thurrot 1999-12-15.) Tämän jälkeen on julkaistu Windows Server 2003, 2003 R2, 2008, 2008 R2, 2012 sekä Windows Server 2012 R2.

Nykyaikaiset palvelinkäyttöjärjestelmät ovat roolipohjaisia, mikä tarkoittaa sitä, että käyttöjärjestelmä asennetaan tyhjänä, minkä jälkeen siihen liitetään halutut roolit eli päätetään, mihin käyttötarkoitukseen palvelinta käytetään. Palvelin voidaan määritellä toimimaan esimerkiksi DHCP-palvelimenä eli lähiverkkoon IP-osoitteita jakavana palvelimenä liittämällä siihen DHCP-palvelinrooli. (Kuva 2.)



KUVA 2. Windows Server 2012 R2 -palvelinroolit (Kauppinen 2015-03-10.)

## 4.2 Käytettävät palvelinversiot

Tässä työssä käytettyjen Windows-palvelinten versiot olivat Windows Server 2008, Windows Server 2008 R2, Windows Server 2012 sekä Windows Server 2012 R2. Lisäksi sähköpostipalvelimina käytettiin Microsoft Exchange Server 2003, 2010 ja 2013 sekä Lync-palvelimina Microsoft Lync Server 2010 ja 2013.

### 4.2.1 Windows Server 2008

Windows Server 2008 julkaistiin 27.2.2008. Se perustui Windows NT 6.0 -käyttöjärjestelmään samoin kuin Windows Vista, joten se sisälsi samat arkkitehtuuriset uudistukset ja parannukset. Windows Server 2008:sta on olemassa useampi eri versio kuten Standard, Enterprise ja Datacenter. Versioiden välisiä eroja ovat muun muassa tuettavat laitteistot ja palvelinroolit sekä käyttökohteet ja hinta. Standard-versio on tarkoitettu pienille - ja keskisuurille yrityksille, Enterprise suurille yrityksille ja Datacenter suurten organisaatioiden käyttöön. (Microsoft 2014a.)

Windows Server 2008:aan ei ole integroituna automaattisesti PowerShellä, vaan se on mahdollista asentaa erikseen. Windows Server 2008:aa julkaistaessa uusin PowerShell versio oli 1.0.

### 4.2.2 Windows Server 2008 R2

Windows Server 2008 R2 julkaistiin 22.7.2009. Vaikka Windows Server 2008 R2 julkaistiin itsenäisenä palvelinkäyttöjärjestelmänä, se on silti enemmän päivitys edeltäjäänsä kuin kokonaan uusi käyttöjärjestelmä. Windows Server 2008 R2 sisälsi monia uusia ja päivitettyjä ominaisuuksia, kuten uuden version IIS web -palvelimesta ja PowerShellistä sekä DirectAccess-toiminnon. (Microsoft 2010.)

Windows Server 2008 R2 -version oli integroituna PowerShell versio 2.0, joka sisälsi aiempaan versioon nähden monia uusia ominaisuuksia, kuten yli 100 uutta cmdlets-komentoa, etäkäyttömahdollisuuden sekä integroidun skriptaus-ympäristön (ISE). (Microsoft 2009.)

### 4.2.3 Windows Server 2012

Windows Server 2012 oli Microsoftin kuudes Windows Server -julkaisu. Se julkaistiin 1.8.2012. Windows Server 2012 perustui Windows NT 6.2 -käyttöjärjestelmään ja se oli käyttöliittymältään samankaltainen kuin Windows 8. Windows Server 2012 suurimpia muutoksia aiempiin versioihin nähden olivat käyttöliittymän lisäksi muun muassa Hyper-V Replikointi, Tallennustilat-työkalu sekä PowerShell versio 3.0. (Microsoft 2014b.)

Windows Server 2012:sta on olemassa neljä eri versiota: Standard, Datacenter, Essentials ja Foundation. Standard- ja Datacenter -versiot ovat samoja tuotteita, joiden erot ovat ainoastaan virtualisoinnissa. Standard versiossa virtuaalipalvelimia voi olla maksimissaan kaksi kappaletta, kun taas

Datacenter versiossa rajoittamattomasti. Nämä versiot on tarkoitettu ensisijaisesti yli 25 työaseman ympäristöihin. Essentials ja Foundation versiot on tarkoitettu pienyrityksille. Näissä versioissa on useita rajoitteita, kuten virtualisoinnin puute sekä rajatut asiakasmäärät. (Lowe 2012-07-06.)

#### 4.2.4 Windows Server 2012 R2

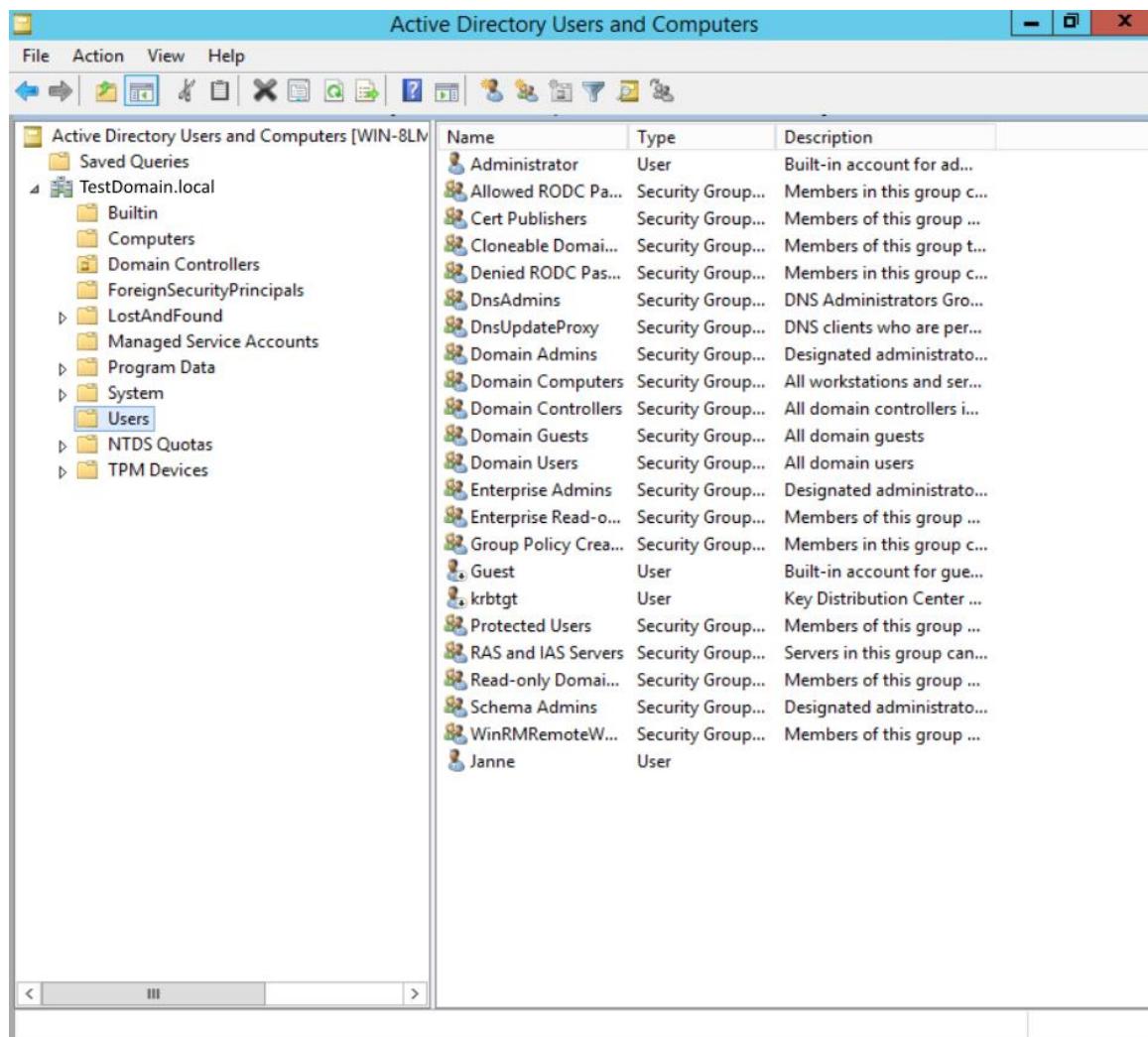
Windows Server 2012 R2 on viimeisin julkaisu Windows Server -tuoteperheestä. Se julkaistiin 18.10.2013. Windows Server 2012 R2 on ensimmäinen Microsoft Server -käyttöjärjestelmä, jolla on liitettävyyttä pilveen. Se sisältää myös viimeisimmän version PowerShellistä sekä yli 300 uutta tai päivitettyä ominaisuutta. Käyttöliittymältään Windows Server 2012 R2 samankaltainen kuin edellinen Windows Server 2012. (Minasi, Greene, Booth, Butler, McCabe, Panek, Rice ja Roth 2014, 1 – 17.)

#### 4.2.5 Microsoft Exchange - ja Lync Server

Microsoft Exchange Server ja Lync Server ovat molemmat Windows Server -käyttöjärjestelmään asennettavia palvelinsovelluksia, joiden avulla voidaan hallita mm. Lync- ja sähköpostitunnuksia sekä sähköpostilaatikoita. Molemmat palvelinsovellukset osaavat hyödyntää Active Directorya objektien hallintaan.

### 4.3 Active Directory

Active Directory (AD) on Microsoftin kehittämä hierarkkinen käyttäjätietokanta ja hakemistopalvelu, joka on integroitu osaksi Windows Server-palvelinympäristöä (kuva 3). Sen avulla voidaan hallita muun muassa organisaation työasemia, Windows-käyttäjätietoja, Exchange-sähköpostitietoja sekä käyttöoikeuksia. AD pohjautuu LDAP-verkkoprotokollaan ja koostuu objekteista, joita ovat esimerkiksi käyttäjät, ryhmät, työasemat ja tulostimet. (Desmond, Richards, Allen ja Lowe-Norris 2013, 5-9.)



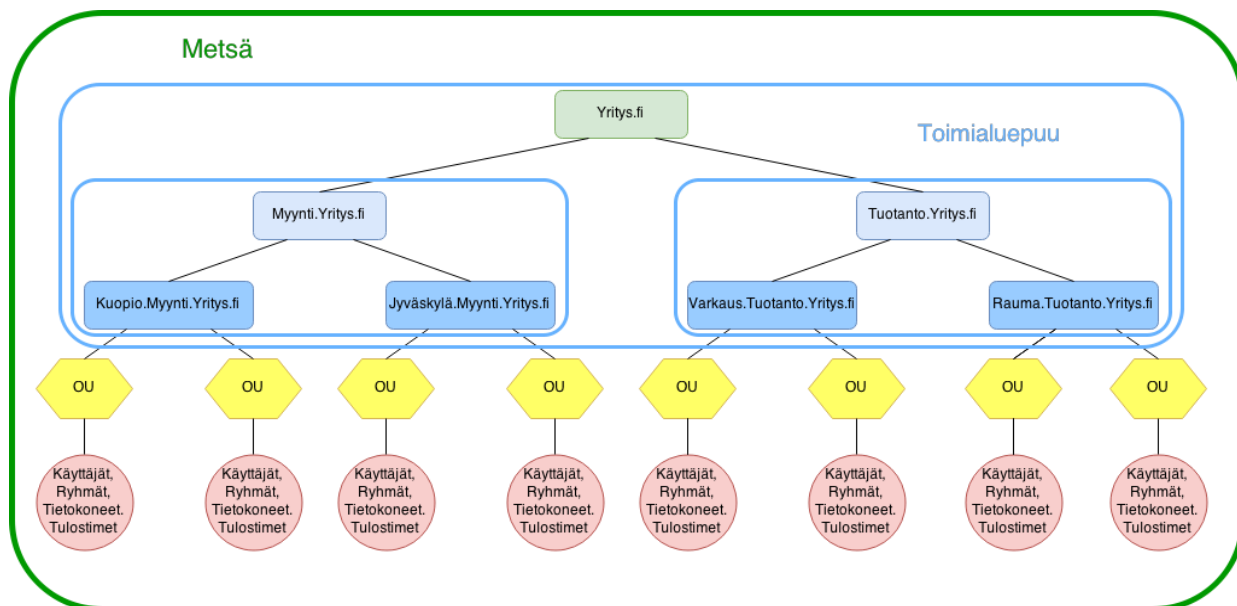
KUVA 3. Windows Server 2012 R2 -Active Directory (Kauppinen 2015-03-19.)

Active Directory on rakenteeltaan hierarkkinen, mikä tarkoittaa sitä, että siinä on useita tasoja. AD:n rakenne määritellään schemassa. Schema sisältää määritelmän jokaisesta objektista, joka voidaan luoda AD -metsään sekä jokaisesta AD -objektin ominaisuudesta. Schema on metsäkohtainen, mikä tarkoittaa sitä, että sen kautta voidaan hallita vain samassa metsässä olevia objekteja ja niiden määrittäjiä. Jokaisella objektilla on oma yksilöllinen GUID-tunnus. (Microsoft 2014c.)

Active Directory rakenne koostuu loogisista ja fyysisistä osista. Loogisia rakenteita ovat toimialue-metsät (forest), toimialuepuut (tree), toimialueet (domain) sekä organisaatioyksiköt (organization unit) ja fyysisiä rakenteita toimipaikat (sites), aliverkot sekä ohjauspalvelimet (domain controller).

Toimialuepuut ovat toimialueita ja alitoimialueita sisältäviä ryhmiä. Toimialuemetsät, eli juuritoimialueet, ovat joukko toimialuepuita. Juuritoimialue nimetään usein yrityksen mukaan, esimerkiksi Savonia.fi, ja se määrittää myös metsän nimen. Kun ensimmäinen toimialue luodaan, tulee siitä myös metsän juuritoimialue. Toimialueet ovat joukko työasemia, jotka käyttävät samaa AD-hakemistoa. Ne määrittelevät organisaation loogisen rakenteen ja mahdollistavat kaikkien siihen kuuluvien objektien hallinnan yhdellä kertaa. Organisaatioyksiköt (OU) ovat hakemistoja, joiden avulla toimialueen objektit voidaan ryhmitellä järkevällä tavalla. OU voi sisältää monia eri objekteja, kuten toisia organisaatioyksiköitä, käyttäjiä sekä työasemia. (Kuvio 1.)

Toimipaikat sisältävät yhden tai useamman aliverkon, jonka avulla voidaan määrittää verkon fyysinen rakenne. Toimipaikkojen avulla AD -objektit tietävät missä sijaitsee lähin ohjauspalvelin. Ohjauspalvelimet ovat palvelimia, jotka ylläpitävät siihen liitettyä AD -hakemistoa. On suositeltavaa, että jokaiselle palvelimelle olisi liitettyä vähintään kaksi ohjauspalvelinta, jotta vikatilanteen sattuessa palvelu ei keskeytyisi. (Stanek 2013, 1135 – 1160.)



KUVIO 1. Active Directoryn esimerkkirakenne (Kauppinen 2015-03-19.)



## 5 POWERSHELL

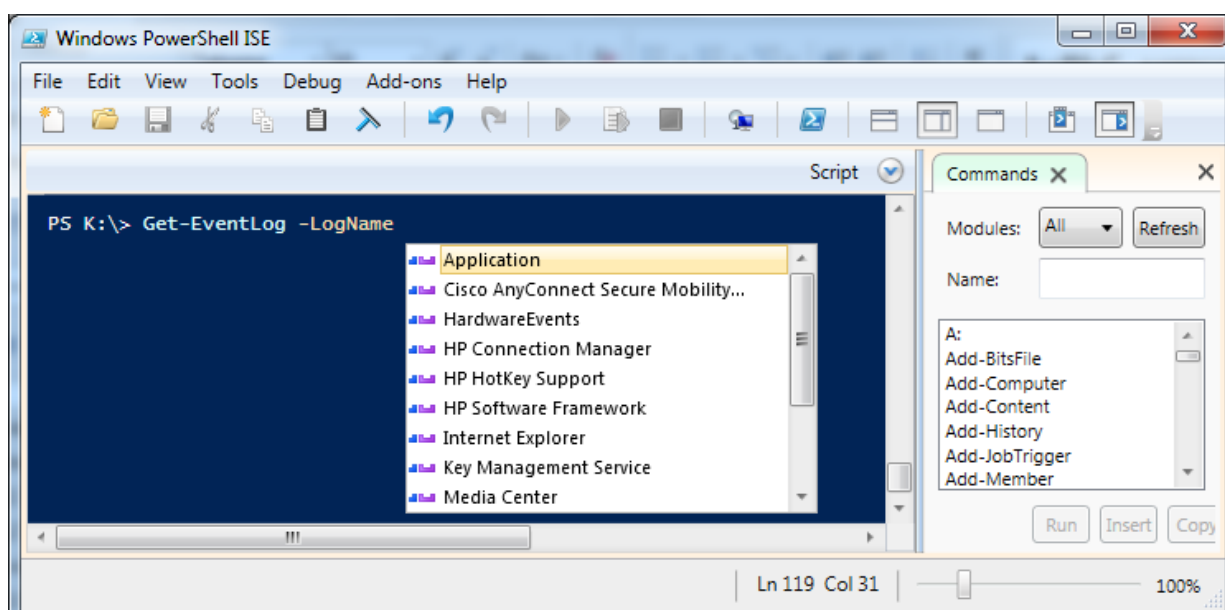
PowerShell on Microsoftin kehittämä oliopohjainen komentotulkki, joka on tarkoitettu ylläpitäjille erilaisten toimintojen hallintaan sekä tehtävien automatisointiin. Se pohjautuu .NET-ohjelmistokehykseen. (Falon 2014-03-19.)

### 5.1 Versiot

PowerShell 1.0 julkaistiin vuonna 2006 Windows XP SP2, Windows Vista sekä Windows Server 2003 -alustoille. Se sisältää 129 eri cmdlets-komentoa. (Microsoft 2014d.)

Seuraava versio, PowerShell 2.0, julkaistiin lokakuussa 2009. Se on integroitu osaksi Windows 7 ja Windows Server 2008 R2 -käyttöjärjestelmiä. PowerShell 2.0 sisältää satoja uusia cmdlets-komentoja, uusia muuttujia sekä integroidun skriptaus-ympäristön (ISE). Lisäksi se mahdollistaa komentojen suorittamisen myös etätyöasemille. (Microsoft 2015a.)

PowerShell 3.0 julkaistiin vuonna 2012 Windows 8- ja Windows Server 2012 R2 -alustalle. Näkyvimpiä uudistuksia edelliseen versioon verrattuna ovat sadat uudet cmdlets-komennot, Power Shell Web Access sekä automaattitäydennys (kuva 4). (Microsoft 2014e.)



KUVA 4. PowerShell ISE -skriptien automaattitäydennys (Kauppinen 2015-03-15.)

PowerShell 4.0 on esiasennettuna Windows 8.1 - sekä Windows Server 2012 R2 -käyttöjärjestelmiin. Se tarjoaa monia päivitettyjä ja korjattuja ominaisuuksia, uusia cmdlets-komentoja sekä uutena ominaisuutena Windows PowerShell Desired State Configuration (DSC) -hallintajärjestelmän. DSC mahdollistaa sovelluspalveluiden sekä ympäristön hallinnan ja käyttöönoton. (Microsoft 2014f.)

Microsoft on julkaissut myös PowerShell 5.0 -esikatseluversion, joka on vielä kehitysasteella. (Microsoft 2014g.)

## 5.2 Komennot

PowerShell-komennot voidaan jakaa neljään eri kategoriaan: cmdlet-komennot, shell-funktiokomennot, skriptikomennot sekä natiivit Windows komennot (Payette 2011, 42).

### 5.2.1 Cmdlet-komennot

Cmdlet-komennot koostuvat verbistä ja substantiivista, esimerkiksi *Get-Volume*, jossa verbi määrittää toiminnon ja substantiivi käsiteltävän objektin (Payette 2011, 13). Kyseinen cmdlet-komento tulostaa työaseman kovalevyjen tiedot. Näille komennoille voidaan asettaa myös parametreja, kuten esimerkiksi *Get-Volume C*, joka tulostaa ainoastaan C-aseman tiedot. Cmdlet-komennot pohjautuvat .NET-ohjelmistokehykseen ja periytyvät cmdlet-luokasta. Cmdlet-komennoilla voidaan esimerkiksi luoda uusia käyttäjiä, poistaa tunnuksia ja muokata rekisteritiedostoja. (Payette 2011, 42 - 43.) Cmdlet-komentoja voidaan käyttää myös aliasten kautta. Alias on pikakomento toiselle komennolle. Kuvassa 5 on suoritettu *dir*-komento, joka listaa työkansion tiedostot. *Get-Command dir*-komento kuitenkin paljastaa, että *dir*-komento onkin alias *Get-ChildItem*-komennolle.

```
PS D:\PowerShell> dir

Directory: D:\PowerShell

Mode                LastWriteTime         Length Name
----                -
-a---             27.3.2015      16:17         89 GetVolume.ps1
-a---             29.3.2015      18:41         17 TextFile.txt

PS D:\PowerShell> Get-Command dir

CommandType      Name
-----
Alias            dir -> Get-ChildItem

PS D:\PowerShell> Get-Command Get-ChildItem

CommandType      Name
-----
Cmdlet           Get-ChildItem
ModuleName       Microsoft.PowerShell.Management

PS D:\PowerShell> Get-ChildItem

Directory: D:\PowerShell

Mode                LastWriteTime         Length Name
----                -
-a---             27.3.2015      16:17         89 GetVolume.ps1
-a---             29.3.2015      18:41         17 TextFile.txt
```

KUVA 5. PowerShell *dir*-alias (Kauppinen 2015-03-26.)

### 5.2.2 Shell-Funktiokomennot

Shell-Funktiokomennot ovat ohjelmapätkiä, joita kutsutaan niiden nimellä. Funktioita käytetään, jotta samaa asiaa ei tarvitsisi toistaa useaan kertaan. Kuvassa 6 on luotu funktio *GetVolume*, joka ottaa parametrina (*\$drive\_letter*) aseman tunnuksen ja palauttaa halutun aseman tiedot. Funktiot pysyvät käytössä istunnon ajan, mikäli niitä ei tallenneta tiedostoon. (Payette 2011, 43.)

```
PS D:\PowerShell> function GetVolume($drive_letter) {Get-Volume $drive_letter}
PS D:\PowerShell> GetVolume C
```

DriveLetter	FileSystemLabel	FileSystem	DriveType	HealthStatus	SizeRemaining	Size
C		NTFS	Fixed	Healthy	37.93 GB	232.37 GB

KUVA 6. PowerShell GetVolume -funktio (Kauppinen 2015-03-26.)

### 5.2.3 Skriptikomennot

Skriptikomennot ovat tiedostoihin tallennettuja komentoja. Ne ladataan ja käännetään jokaisella suorituskerralla, mikä tekee niistä jonkin verran hitaampia käynnistymään kuin funktiot. Skriptitiedostot ovat .ps1-päätteisiä. (Payette 2011, 44.) Kuvassa 7 nähdään, että GetVolume.ps1-tiedostoon on tallennettu kuvan 6 *GetVolume* -funktio, joka ensiksi ladataan käytettäväksi ja tämän jälkeen suoritetaan.

```
GetVolume.ps1 X
1 #GetVolume.ps1 - Script
2
3 Function GetVolume($drive_letter){Get-Volume $drive_letter}
```

```
PS D:\PowerShell> . .\GetVolume.ps1
PS D:\PowerShell> getvolume -drive_letter C
```

DriveLetter	FileSystemLabel	FileSystem	DriveType	HealthStatus	SizeRemaining	Size
C		NTFS	Fixed	Healthy	37.93 GB	232.37 GB

KUVA 7. PowerShell GetVolume -skripti (Kauppinen 2015-03-26.)

### 5.2.4 Natiivit Windows-komennot

Natiivikomennot ovat ulkopuolisia ohjelmia, jotka voidaan suorittaa käyttöjärjestelmästä. Natiivikomennot ovat hitaimpia komentotyyppejä, koska niiden suorittaminen vaatii kokonaan uuden prosessin perustamisen. (Payette 2011, 44 - 46.)

### 5.3 Putkitus

PowerShell-komentoja voidaan putkittaa keskenään eli yhdistää useampi komento yhdeksi komentosarjaksi. Komennot putkitetaan käyttämällä "|" -merkkiä. Esimerkiksi kuvassa 8 näkyvä komentoputki "*Get-Process | Where Name -Match "Service" | Sort-Object Name -Descending*" on putkitettu käyttäen kolmea komentoa: *Get-Process* -komento hakee käynnissä olevat prosessit, *Where Name -Match "Service"* -komento asettaa *Get-Process*-komennolle hakuehdon, missä prosessin nimen on sisällettävä sana "Service", ja *Sort-Object Name -Descending* -komento järjestee prosessit nimen perusteella laskevasti.

```
PS D:\PowerShell> Get-Process | where Name -Match "Service" | Sort-Object Name -Descending
```

Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
403	38	48784	5364	221		9204	XtuService
463	27	6832	9900	99		2808	TeamViewer_Service
294	10	5836	6380	30		712	services
106	10	1420	2900	47		2408	NvNetworkService
97	6	944	788	49		4112	ngservice
124	8	1868	2424	59		2344	GfExperienceService
367	24	24508	15876	172		2688	GameScannerService
197	11	4064	5648	61		2224	AsusFanControlService

KUVA 8. PowerShell-komentojen putkitus (Kauppinen 2015-03-27.)

Komentoputken komentojen suorituserjestyä voidaan muuttaa käyttämällä sulkeita (Payette 2011, 60 - 63).

## 6 TEKNINEN TOTEUTUS

### 6.1 Lähtökohdat ja vaatimukset

Service Deskissä on käytössä Microsoft-DOS-komentoja sisältäviä .BAT-skriptitiedostoja, Visual Basic -skriptejä sekä PowerShell-skriptejä. Vaikka Visual Basic:llä pystytään tekemään samoja asioita (esim. käyttäjätunnusten luonti) kuin PowerShellillä, PowerShell tarjoaa tähän yksinkertaisemman vaihtoehdon Windows-ympäristössä. Tästä syystä työ päätettiin toteuttaa PowerShellillä.

Service Deskissä aiemmin käytetyt PowerShell-skriptit on tehty pääpiirteittäin asiakaskohtaisiksi, mikä tarkoittaa sitä, että niiden hyödyntäminen toisilla asiakkailla ei onnistu ilman skriptin laajempaa muokkaamista. Tämän takia skriptien vaatimuksiksi asetettiin dynaamisuus sekä käyttöönoton helpous jatkossa eri asiakkuuksissa.

### 6.2 Suunnittelu

Työn suunnittelu aloitettiin tutustumalla PowerShellin käyttömahdollisuuksiin sekä selvittämällä sen hetkiset PowerShellin käyttökohteet ja käyttötarpeet Service Deskissä. Selvityksen perusteella päätettiin, mitä skriptejä ja mihin tarkoitukseen niitä luodaan.

Työtä varten luotiin virtuaalikone, jonka käyttöjärjestelmäksi asennettiin Windows Server 2012 R2. Virtuaalipalvelimelle asennettiin Active Directory Domain Services -rooli, mikä mahdollistaa AD:n käytön, sekä Windows Exchange Server 2013 -sähköpostipalvelin sähköpostilaatikoiden luontia varten. Virtuaalinen testiympäristö mahdollisti riippumattomuuden Enfon ja sen asiakasyritysten palvelimista ja sen avulla voitiin luoda ja testata skriptejä eristetyssä ympäristössä.

### 6.3 Toteutus ja testaus

Skriptit pyrittiin kokoamaan mahdollisimman monesta järkevästä funktiosta, jotta samaa asiaa tai toimintoa ei tarvitsisi kirjoittaa aina uudestaan. Esimerkiksi ääkkösten muuttaminen toteutettiin omassa funktiossa erillisessä tiedostossa, jolloin sen käyttämiseksi ei tarvinnut muuta kuin sisällyttää funktio skriptiin komennolla `". Sijainti\FunktionNimi.ps1"` ja kutsua funktiota sen vaatimilla parametreilla: `"FunktionNimi $param1 $param2"`. Osassa skripteistä on hyödynnettiin toimipaikan ohjaajan Tomi Monosen aiemmin luotuja skriptejä.

Suurimmat PowerShellin hyödyntämistarpeet Service Deskissä liittyvät erilaisten tunnusten, ryhmien ja sähköpostilaatikoiden sulkemiseen ja poistamiseen sekä luomiseen ja muokkaamiseen. Eri asiakkuuksilla on omat menetelmänsä tehdä nämä toiminnot, minkä takia saman kokonaisuuden toteuttamiseen on useita eri variaatioita.

### 6.3.1 Tunnusten poistaminen

Esimerkiksi asiakkaalla X käyttäjä poistetaan seuraavalla tavalla:

1. Etsitään poistettava henkilö yrityksen AD:sta.
2. Kirjoitetaan poistettavan henkilön ryhmät talteen.
3. Nimetään henkilön kotihakemisto uudelleen määrättyllä etuliitteellä.
4. Poistetaan tunnus.

Kyseisessä tapauksessa skripti toteutettiin kysymällä ensiksi poistettavan käyttäjän tunnus ja tarkistamalla, että annettu tunnus on olemassa. Tämän jälkeen kutsuttiin funktiota, joka kirjoittaa käyttäjän ryhmät tekstitiedostoon. Seuraavaksi kutsuttiin funktiota, joka nimeää kotihakemiston määrättyllä etuliitteellä, ja lopuksi vietiin poistettava tunnus funktioon, joka poistaa sen järjestelmästä.

(Kuva 9.)

```
Import-Module ActiveDirectory
$sp = Split-Path -Parent $PSCmdPath #Scriptien juurikansio

. $sp\NimeaKotihakemisto.ps1 #Ottaa tunnuksen, palauttaa true/false
. $sp\RyhmätTiedostoon.ps1 #Ottaa tunnuksen, palauttaa true/false
. $sp\PoistaTunnus.ps1 #Ottaa tunnuksen

$henk = Read-Host "Anna käyttäjän tunnus: "
$user = Get-ADUser -f {samaccountname -eq $henk}

while($user -eq $null) {
    $henk = Read-Host "Käyttäjää ei löytynyt. Anna käyttäjän tunnus: "
    $user = Get-ADUser -f {samaccountname -eq $henk}
}

$rt = RyhmätTiedostoon($henk)
if($rt -eq $true) { Write-Host "Käyttäjän $henk ryhmät tallennettu tiedostoon $sp\Ryhmät\$henk.txt." }

$nhk = NimeaKotihakemisto($henk)
if($nhk -eq $true) { Write-Host "Käyttäjän $henk kotihakemisto nimetty uudelleen." }
else { Write-Host "Käyttäjän $henk kotihakemistoa ei nimetty uudelleen, koska sitä ei löytynyt." }

PoistaTunnus($henk)
Write-Host "Käyttäjä $henk poistettu."
```

KUVA 9. Käyttäjän poistaminen Active Directorystä (Kauppinen 2015-04-15.)

Asiakas X:n tunnusten poisto -skriptiä varten kirjastoon luotiin seuraavat funktiot, joita voidaan hyödyntää jatkossa uusissa skripteissä:

- NimeaKotihakemisto.ps1 – Nimeää kotihakemiston määrättyllä etuliitteellä.
- RyhmätTiedostoon.ps1 – Tallentaa käyttäjän käyttäjäryhmät tekstitiedostoon.
- PoistaTunnus.ps1 – Poistaa tunnuksen.

Kuten aiemmin todettiin, eri asiakkailla on erilaisia variaatioita, kuinka tunnus poistetaan. Esimerkki-asiakkaalla Y käyttäjän poisto tehdään alla olevalla tavalla:

1. Etsitään poistettava henkilö yrityksen AD:sta.
2. Poistetaan kotikansio.
3. Poistetaan virtuaalisen työpöytäympäristön luomat profiilikansiot.
4. Poistetaan tunnus.

Asiakas Y:n tunnusten poisto -skriptiin hyödynnettiin aiemmin luotua asiakas X:n tunnusten poisto -skriptiä pienin muokkauksin. Kun käyttäjätunnus on kysytty, etsitään käyttäjän kotihakemisto ja poistetaan se. Tätä varten kirjastoon luotiin uusi funktio "PoistaKansio", joka ottaa parametrinä poistettavan kansion polun, tarkistaa, että kansio on olemassa, sekä poistaa kansion ja palauttaa arvon "True" tai "False". Tämän jälkeen samaan funktioon viedään virtuaalisen työpöytäympäristön profiilikansioiden polku ja poistetaan pyydytetyt kansiot. Lopuksi poistetaan käyttäjä viemällä tunnus "PoistaTunnus" -funktioon.

Mikäli käyttäjätunnusta ei tahdota poistaa esimerkiksi työntekijän pitkän loman takia, se voidaan sulkea komennolla "*Get-ADUser \$tunnus / Disable-ADAccount*". Tällöin työntekijän palatessa takaisin töihin voidaan tunnus ottaa uudelleen käyttöön.

### 6.3.2 Tunnusten luonti

Tunnusten luonti on iso osa Service Desk -työntekijän työtä. Kun hallitaan useiden asiakasyritysten käyttäjiä, voi tunnusten luonnit täyttää koko työpäivän. Tunnuksia luotaessa joudutaan usein ottamaan etäyhteys useisiin eri palvelimiin, kuten AD-palvelimeen, sähköpostipalvelimeen sekä Lync-palvelimeen, mikä hidastaa entisestään työntekoa.

Myös tunnusten luonnissa on asiakaskohtaisia eroja. Osa asiakkaista lähettää tunnustilaukset ilmoittamalla henkilön nimen ja toimipaikan, osa oman järjestelmän kautta luodulla lomakkeella, kun taas osa ilmoittaa käyttäjän perustiedot ja tunnuksen, mistä loput käyttäjä- ja käyttöoikeustiedot voidaan kopioida. Näitä eri vaihtoehtoja varten luotiin skriptit, joiden avulla uuden tunnuksen luonti saadaan tehtyä yhden palvelimen kautta mahdollisimman tehokkaasti.

#### 6.3.2.1 Uuden käyttäjän luonti nimen ja toimipaikan perusteella

Uuden käyttäjätunnuksen luonti nimen ja toimipaikan perusteella toteutettiin seuraavalla tavalla:

1. Kysytään uuden käyttäjän nimi ja toimipaikka.
2. Etsitään oikea OU toimipaikan perusteella.
3. Luodaan käyttäjä Active Directoryyn.
4. Kirjoitetaan uuden käyttäjän tunnus ja salasana tekstitiedostoon.
5. Lisätään käyttäjäryhmät.
6. Muokataan käyttäjän kotihakemiston käyttöoikeudet.
7. Luodaan sähköpostitunnus, mikäli tunnustilauksessa se on pyydetty.
8. Luodaan Lync-tunnus, mikäli tunnustilauksessa se on pyydetty.
9. Luodaan käyttäjä Enfon MyEnfo-asiakasjärjestelmään.

Uuden käyttäjän luonti Active Directoryyn tehtiin kutsumalla funktiota "*Luo\_Tunnus*" parametreilla etunimi, sukunimi, toimipaikka ja OU sekä antamalla määritetyistä parametreista koottu avain-arvo-paritaulu parametrina Cmdlet-komennolle "*New-ADUser*" (kuva 10).

```

$params = @{
    "AccountPassword"=$password_ss
    "ChangePasswordAtLogon"=$true
    "DisplayName"= $kokonimi
    "Enabled"=$true
    "GivenName"=$enimi
    "Homedrive" = "H:"
    "HomeDirectory" = "\\polku\$sam"
    "Name"=$kokonimi
    "Office" = $tpaikka
    "Path"=$ou
    "SurName"=$snimi
    "SamAccountName" = $sam
    "UserPrincipalName"= "$sam@$upn_end"
}

New-ADUser @params

```

KUVA 10. Uuden käyttäjän luonti Active Directoryyn *New-ADUser* -komennolla (Kauppinen 2015-04-19.)

Kaikille uusille käyttäjille lisättävien käyttäjäryhmien nimet kovakoodattiin muuttujiin ja vietiin parametreina funktioon *"Lisaa\_Ryhma"*. Ryhmät lisättiin komennolla:

*Add-ADGroupMember \$RyhmänNimi -Members \$KäyttäjäTunnus*

Mikäli tunnustilauksessa pyydetään lisäämään uudelle käyttäjälle muita ryhmiä, voidaan ne lisätä antamalla ryhmän nimi sitä kysyttäessä (Kuva 11).

```

Lisätäänkö käyttäjälle muita ryhmiä (k/e)?
k
Anna ryhmän nimi: Ryhmä1
Käyttäjä lisättiin ryhmään Ryhmä1
Lisätäänkö käyttäjälle muita ryhmiä (k/e)?
e

```

KUVA 111. Käyttäjän lisäys käyttäjäryhmiin (Kauppinen 2015-04-19.)

Uuden käyttäjän kotihakemiston käyttöoikeuksista poistettiin oikeuksien periytyminen kotihakemiston juurihakemistosta kotihakemistoon ja sen alikansioihin sekä lisättiin hakemiston täydet käyttöoikeudet uudelle käyttäjälle sekä järjestelmänvalvojalle. Tätä varten luotiin funktio *"Muokkaa\_Hakemistooikeuksia"* mikä ottaa vastaan parametrina käyttäjätunnuksen. Funktio lisää ensiksi käyttöoikeuden halutulle henkilölle ja tämän jälkeen poistaa kansio-oikeuksien periytyminen. Lopuksi funktio poistaa kansio-oikeudet ylimääräisiltä käyttäjiltä jättäen käyttöoikeuden vain järjestelmänvalvojalle sekä halutulle henkilölle.



```

try{
    $acl = Get-Acl -path $kansio
    $kansio = (convert-path $acl.pspath)
    Foreach($access in $acl.access)
    {
        Foreach($value in $access.identityReference.Value)
        {
            if (!($value -eq $user) -and !($value -eq $admin))
            {
                $acl.RemoveAccessRule($access) | Out-Null
            }
        }
    }
    Set-Acl -path $kansio -aclObject $acl
    Write-Host "Ylimääräiset käyttöoikeudet poistettu." -ForegroundColor Green
} catch { Write-Host "Ylimääräisiä käyttöoikeuksia ei poistettu." -ForegroundColor Red}

```

KUVA 12. Käyttöoikeuksien poistaminen kansioista (Kauppinen 2015-04-22.)

Kuvassa 12 on esitetty kuinka käyttöoikeudet saadaan poistettua halutuilta henkilöiltä tietystä kansioista. Oikeuksien poistaminen suoritetaan *try-catch*-lauseella, jolloin virheisiin päästään helposti käsi. "*Get-Acl*"-komennolla haetaan kirjastomuuttujaan kansion tämänhetkiset oikeudet. Tämän jälkeen käydään jokainen oikeus yksi kerrallaan läpi ja verrataan yksittäiseen oikeus-objektiin sidottua käyttäjää ennalta määritettyihin käyttäjiin. Mikäli objektiin sidottu käyttäjä ei vastaa ennalta määritettyjä käyttäjiä, poistetaan se kirjastosta. Kun koko kirjasto on käyty läpi ja ylimääräiset oikeudet on poistettu, asetetaan kansiolle kirjaston mukaiset oikeudet "*Set-Acl*"-komennolla.

Uuden käyttäjän luonti nimen ja toimipaikan perusteella -skripti luotiin suoritettavaksi sähköpostipalvelimelta, mutta mikäli asiakkaan tunnusmäärityksissä olisi määritelty, että AD-tunnus ja sähköpostitunnus luodaan omilta palvelimiltaan, voitaisiin yhteys ottaa ennen sähköpostin luomista sähköpostipalvelimelle seuraavalla komennolla:

```
Enter-PSSession -ComputerName $PalvelimenNimi -Credential $Tunnus
```

Sähköposti aktivoitiin funktiossa "*Aktivoi\_Spost*" joka ottaa parametrina käyttäjäobjektin.

Lync-tunnuksien luonti tehtiin funktiossa "*Aktivoi\_Lync*", jonne vietiin parametrina käyttäjätunnus. Koska AD-käyttäjätunnus sekä sähköposti oli luotu ennen lync-tunnusten luontia, voitiin lync-tunnukset aktivoida näihin tunnuksiin. Lync-tunnusten aktivointi täytyi suorittaa lync-palvelimelta, johon yhteys saatiin aiemmin mainitulla "*Enter-PSSession*"-komennolla. Tunnusten aktivointi ja määrittely suoritettiin komennolla:

```
Enable-CsUser $Käyttäjä -RegistrarPool $PoolinNimi -SipAddressType EmailAddress | Grant-CsExternalAccessPolicy -PolicyName $PolitiikanNimi
```

Uuden käyttäjän luonti Enfon MyEnfo-asiakasjärjestelmään toteutettiin funktiossa "*Li-saa\_Kayttaja\_Myenfo*" JSON-rajapinnan kautta. Kyseinen rajapinta oli luotu aiemmin toista projektia

varten. *"Lisaa\_Kayttaja\_Myenfo"* -funktio ottaa parametrina käyttäjätunnuksen ja hakee sen perusteella uuden MyEnfo käyttäjän luontiin tarvittavat tiedot AD:sta. Seuraavaksi parametreista muodostetaan JSON-objekti, joka sisältää myös tarvittavan tiedon siitä, mitä parametreilla tehdään. Tämän jälkeen lähetetään HTTPS-pyyntö palvelun verkko-osoitteeseen komennolla:

```
Invoke-RestMethod -Uri $Osoite -Method Post -Credential $TunnusJaSalasana -ContentType "application/json" -Body $JSONObjekti
```

PowerShell muuttaa palvelusta palautuvan vastauksen tyyppin perustuen sinne viedyn datan tyyppiin. Tässä tapauksessa PowerShell muuttaa palautuvan vastauksen objektiksi, jonka arvoihin päästään käsiksi komennolla:

```
$muuttuja.records[0].HaluttuArvoNimi
```

Mikäli haluttaisiin päästä käsiksi esimerkiksi tietoon, onnistuiko uuden käyttäjän luonti, niin halutun arvon nimeksi muutettaisiin *"\_\_status"*. Lopuksi, kun vastaus on käsitelty, ilmoitetaan onnistuiko uuden käyttäjän lisäys.

#### 6.3.2.2 Uuden käyttäjän luonti lomakepohjasta

Osalla asiakasyrityksistä on omat järjestelmät, joista voidaan muun muassa ilmoittaa uusista työntekijöistä sekä tilata tunnuksia, sähköpostilaatikoita ja käyttöoikeuksia. Kun asiakas ilmoittaa uudesta käyttäjästä järjestelmänsä kautta, muodostuu siitä Enfon tikettijärjestelmään uusi työpyyntö. Asiakkaan järjestelmästä lähetetty työpyyntö on normaalisti tietyn lomakepohjan mukainen (Kuva 13).

```
New Employee
=====

Uuden työntekijän käyttäjätiedot
-----

Etunimi: Testi
Sukunimi: Testaaja
Puhelinnro: 0501234567
Työsuhteen kesto: Toistaiseksi
Työsuhde alkaa: 01/04/2015
Työsuhde päättyy: 31/12/2015
Esimies: Esko Esimies, Eesimies@yritys.fi
Yhtiö: Yritys Oy
Osasto: Henkilöstöhallinta
Toimipiste: Yritys Oy, Yritystie 1, 01234 Yritys
Sijainti (kerros/huone): 1 kerros
Työntekijän status: Henkilökunta
Ulkoisen työntekijän tarkennus:
Luodaan s-postiosoite: Kyllä
Kustannuspaikka: 1700
Lisätiedot (tarvittaessa):
```

KUVA 13. Esimerkki uuden käyttäjän tilauslomakkeesta (Kauppinen 2015-04-16.)

Esimerkki asiakkaalla X tunnusten luonti -prosessi on seuraava:

1. Tarkista ettei tunnus ole jo käytössä.
2. Määritä käyttäjätiedot ja luo tunnus oikeaan AD-haaraan.
3. Pyydettyessä luo käyttäjälle uusi sähköpostilaatikko.
4. Lisää käyttäjä oikeisiin käyttäjäryhmiin.

Käyttäjän luonti lomakepohjasta toteutettiin siten, että Service Desk työntekijän tarvitsee vain kopioida tilauslomake tekstitiedostoon, suorittaa skripti ja tarvittaessa antaa pyydettyä parametri. Tällä tavoin työntekijän tekemät virheet saadaan minimoitua ja työaika käytettyä tehokkaasti.

Koska tunnustilaus tehdään valmiiksi jäsennetyllä lomakkeella, on se helppo myös parametrisoida, eli muuttaa "*muuttuja=arvo*" -muotoon. Tässä tapauksessa tekstitiedoston arvot tallennettiin tauluun avain–arvo-pareiksi (Kuva 14). PowerShellissä avain–arvo-pari taulua kutsutaan *hashtableksi*. Hashtablen avulla funktioiden välinen parametrien siirto saatiin yksinkertaistettua ja lomakkeen tietojen käsittely helpottui. Esimerkiksi lomakkeella ilmoitettuun sukunimen arvoon päästiin käsiksi hashtable-muuttujan indeksillä "Sukunimi".

Name	Value
----	----
Sukunimi	Testaaja
Työsuhteenkesto	Toistaiseksi
Etunimi	Testi
Esimies	Esko Esimies, Eesimies@yritys.fi
Yhtiö	Yritys Oy
Työntekijänstatus	Henkilökunta
Luodaans-postiosoite	Kyllä
Lisätiedot(tarvittaessa)	
Sijainti(kerros/huone)	1 kerros
Kustannuspaikka	1700
Työsuhdealkaa	01/04/2015
Puhelinno	0501234567
Toimipiste	Yritys Oy, Yritystie 1, 01234 Yritys
Osasto	Henkilöstöhallinta
Työsuhdepäättyy	31/12/2015
Ulkoisentyöntekijäntarkennus	

KUVA 14. Tilauslomake muutettuna avain–arvo-pari tauluksi (Kauppinen 2015-04-16.)

Kun lomakkeen tiedot on muutettu avain–arvo-pari tauluksi, pyydetään työntekijää valitsemaan luodaanko organisaation sisäinen vai ulkopuolinen tunnus. Näiden kahden tunnustyyppin välillä on eroja muun muassa tunnuksen muodossa sekä organisaatioyksikössä ja sähköpostiasetuksissa.

Tämän jälkeen suoritetaan tarvittavat tarkistukset. Mikäli tarkistuksissa löytyy esimerkiksi tunnus, joka on lähes samankaltainen nyt luotavan tunnuksen kanssa, tulee tästä ilmoitus ja mahdollisuus peruuttaa toiminta.

Tarkistusten jälkeen perustetaan käyttäjä ja kirjoitetaan tunnus sekä salasana tekstitiedostoon. Tietojen tallennus tiedostoon tehdään, jotta tunnusten luonnin jälkeen ne voidaan toimittaa tilaajalle. Lopuksi luodaan sähköpostilaatikko funktiossa "*Aktivoi\_Sposti*" ja lisätään käyttäjäryhmät. Kun käyttäjätunnus on luotu ennen sähköpostin luomista, voidaan sähköposti aktivoida komennolla:

*Enable-Mailbox -Identity '\$OU/\$sukunimi \$etunimi' -Alias "\$tunnus" -Database "\$SpostiTietokanta"*

### 6.3.2.3 Uuden käyttäjän luonti kopiosta

Uuden käyttäjän luonti kopioimalla tiedot toisesta tunnuksesta on yksinkertainen tehdä ilman PowerShell-skriptejäkin, mutta se voi olla hyvinkin aikaa vievää, mikäli lisättäviä käyttäjäryhmiä on erityisen paljon.

Käyttäjän luonti kopiosta -skripti vaatii uuden käyttäjän etunimen, sukunimen sekä kopioitavan tunnuksen. Kun vaadittavat arvot on annettu, kutsutaan "*Kopioi\_Kayttaja*"-funktiota, jossa tehdään tarvittavat tarkistukset ja luodaan uusi käyttäjä samaan OU:hun kopioimalla tiedot toisesta käyttäjätunnuksesta. Tämän jälkeen uusi käyttäjätunnus ja salasana kirjoitetaan tekstitiedostoon. Lopuksi kopioidaan uusi käyttäjä vastaaviin käyttäjäryhmiin kuin kopioitava tunnuskin.

### 6.3.3 Sähköpostilaatikon luonti resursseille

Yrityksissä tulee usein vastaan tilanteita, joissa jollekin resurssille tarvitaan oma sähköpostilaatikkonsa. Tällaisia resursseja voivat olla muun muassa neuvotteluhuoneet ja laitteet. Luomalla tällaisen resurssin käyttäjät voivat tehdä siihen varauksia oman Outlook-sovelluksen kautta.

Muutamille Enfon asiakasyrityksille oli jo olemassa PowerShell-skripti tällaisen resurssipostilaatikon luontiin, joten päätin muokata näistä version, jonka voisi ottaa tarvittaessa käyttöön myös muilla asiakkuuksilla.

Resurssipostilaatikon luonti -skripti toteutettiin pyytämällä aluksi luontiin vaadittavat parametrit. Tämän jälkeen luotiin uusi AD-käyttäjä ja sähköpostilaatikko "*New-Mailbox*" -komennolla antamalla sille tarvittavat parametrit. "*New-Mailbox*" -komentoa voidaan käyttää, kun AD-käyttäjää ei ole aiemmin luotu. Tämän jälkeen resurssia varten luotiin uusi ryhmä ja lisättiin halutut henkilöt tähän ryhmään. Lopuksi määritettiin vaadittavat oikeudet sekä sähköpostilaatikkoon että ryhmään.

### 6.3.4 Testaus

Skriptit pyrittiin testaamaan omalle työasemalle luodussa virtuaalisessa testiympäristössä mahdollisimman tarkasti ennen tuotantoympäristöön vientiä. Testaus suoritettiin niin todennäköisillä kuin epätodennäköisillä arvoilla, jotta mahdolliset virhetilanteet tuotantoympäristössä saatiin minimoitua. Koska virtuaalisen testiympäristön käyttöjärjestelmänä oli uusien Windows Server -versio, jouduttiin skripteihin tekemään pieniä muutoksia tuotantoympäristön käyttäessä vanhempaa Windows-Server-versiota.

## 7 YHTEENVETO

Opinnäytetyön tavoitteena oli selvittää, mihin käyttötarkoituksiin Enfo Service Deskissä hyödynnetään PowerShellia ja missä sitä voitaisiin jatkossa hyödyntää, ja selvityksen perusteella luoda PowerShell-funktiokirjasto. Työn aikana luotiin skriptejä niin Active Directory -, Exchange - kuin Lync - objektien hallintaan. Lisäksi skriptejä luotiin hakemistojen sekä hakemisto-oikeuksien muokkaamiseen.

Skriptit toteutettiin hyödyntäen Enfo Oyj:n asiakasohjeita, joista selvisi asiakkaan toimintaperiaate jokaista kokonaisuutta luotaessa. Funktiokirjaston kooksi tuli noin 30 valmista funktiota, joita voidaan hyödyntää useimmilla Enfon asiakasyrityksistä, sekä vajaat 10 valmista skripti-kokonaisuutta.

Eri palvelinversioiden välisiin PowerShell-komentojen eroihin olisi ollut hyvä paneutua heti työn alussa tarkemmin, jolloin skriptien käyttöönotto tuotantoympäristössä olisi sujunut nopeammin.

PowerShell tarjoamat jatkokehitysmahdollisuudet ovat laajat, sillä kaikki mitä voidaan tehdä graafisesta käyttöliittymästä, voidaan myös toteuttaa PowerShellillä. Enfo Oyj:n kannalta suurimmat kehittämisen kohteet liittyvät skriptien käyttöönoton laajentamiseen yhä useamman asiakasyrityksen tarpeisiin.

Opinnäytetyön teko oli mielenkiintoinen ja opettavainen kokemus. Koska minulla ei ollut aiempaa kokemusta PowerShellistä, oli työn suorittaminen paikoitellen hyvinkin haastavaa ja vaati paljon asiaan perehtymistä. Opinnäytetyön loppuraportin lisäksi tuotettiin Enfo Oyj:lle myös dokumentti jokaisesta skriptistä ja funktiosta. Dokumentista selviää skriptien ja funktioiden käyttämiseen tarvittavat tiedot. Dokumentti sisältää asiakaskohtaisia tietoja, minkä vuoksi sitä ei voida julkaista opinnäytetyön yhteydessä.

## LÄHTEET

DESMOND, Brian, RICHARDS, Joe, ALLEN, Robbie ja LOWE-NORRIS, Alistair G. 2013. Active Directory. 5. painos. Sebastopol: O'Reilly Media.

ENFO OYJ 2014. Enfo lyhyesti. [verkkajulkaisu]. Enfo Oyj. [Viitattu 2015-03-03.] Saatavissa: <http://www.enfo.fi/enfo-group/enfo-lyhyesti/>

FALON, Vlad 2014-03-19. Simple Questions: What is PowerShell & What Can You Do With It? [verkkajulkaisu]. 7Tutorials. [Viitattu 2015-4-29.] Saatavissa: <http://www.7tutorials.com/simple-questions-what-powershell-what-can-you-do-it>

LOWE, Scott 2012-07-06. Microsoft announces four Windows Server 2012 editions: What you need to know. [verkkosivusto]. TechRepublic. [Viitattu 2015-3-16.] Saatavissa: <http://www.techrepublic.com/blog/data-center/microsoft-announces-four-windows-server-2012-editions-what-you-need-to-know/>

MICROSOFT 2009. What's New in Windows PowerShell. [Viitattu 2015-3-11.] Saatavissa: [https://technet.microsoft.com/en-us/library/dd378784\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd378784(v=ws.10).aspx)

MICROSOFT 2010. Changes in Functionality from Windows Server 2008 to Windows Server 2008 R2. [Viitattu 2015-3-11.] Saatavissa: [https://technet.microsoft.com/en-us/library/dd391932\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/dd391932(v=ws.10).aspx)

MICROSOFT 2014a. Differences Between the Editions of Windows Server 2008. [Viitattu 2015-3-10.] Saatavissa: <http://social.technet.microsoft.com/wiki/contents/articles/3556.differences-between-the-editions-of-windows-server-2008.aspx>

MICROSOFT 2014b. What's New in Windows Server. [Viitattu 2015-3-16.] Saatavissa: <https://technet.microsoft.com/en-us/library/dn250019.aspx>

MICROSOFT 2014c. What Is the Active Directory Schema? [Viitattu 2015-3-19.] Saatavissa: [https://technet.microsoft.com/en-us/library/cc784826\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc784826(v=ws.10).aspx)

MICROSOFT 2014d. PowerShell 1.0 Cmdlets. [Viitattu 2015-3-22.] Saatavissa: <https://social.technet.microsoft.com/wiki/contents/articles/13769.powershell-1-0-cmdlets.aspx>

MICROSOFT 2014e. What's New in Windows PowerShell: New features in Windows PowerShell 3.0. [Viitattu 2015-3-22.] Saatavissa: [https://technet.microsoft.com/en-us/library/hh857339.aspx#BKMK\\_wps3](https://technet.microsoft.com/en-us/library/hh857339.aspx#BKMK_wps3)

MICROSOFT 2014f. What's New in Windows PowerShell: New features in Windows PowerShell 4.0. [Viitattu 2015-3-22.] Saatavissa: [https://technet.microsoft.com/en-us/library/hh857339.aspx#BKMK\\_wps4](https://technet.microsoft.com/en-us/library/hh857339.aspx#BKMK_wps4)

MICROSOFT 2014g. What's New in Windows PowerShell: New features in Windows PowerShell 5.0. [Viitattu 2015-3-22.] Saatavissa: [https://technet.microsoft.com/en-us/library/hh857339.aspx#BKMK\\_new50](https://technet.microsoft.com/en-us/library/hh857339.aspx#BKMK_new50)

MICROSOFT 2015a. Windows Management Framework. [Viitattu 2015-3-16.] Saatavissa: <http://support.microsoft.com/en-us/kb/968929>

MINASI, Mark, GREENE, Kevin, BOOTH, Christian, BUTLER, Robert, MCCABE, John, PANEK, Robert, RICE, Michael ja ROTH, Stefan. 2014. Mastering Windows Server 2012 R2. 1. painos. Indianapolis: John Wiley & Sons Inc.

PAYETTE, Bruce. 2011. Windows PowerShell In Action. 2. painos. New York: Manning Publications Co.

STANEK, William R. 2013. Microsoft Windows Server 2012 Inside Out. 1. painos. Redmond: Microsoft.

THURROT, Paul 1999-12-15. Road to Gold: A Look at the Development of Windows 2000. [verkkoaineisto]. SuperSite for Windows. [Viitattu 2015-3-16.] Saatavissa: <http://winsupersite.com/article/windows-2000/the-road-to-gold-the-development-of-windows-2000-reviewed-127364>